

UNIVERSIDADE FEDERAL DE PELOTAS

Curso de Ciência da computação



Análise de desempenho

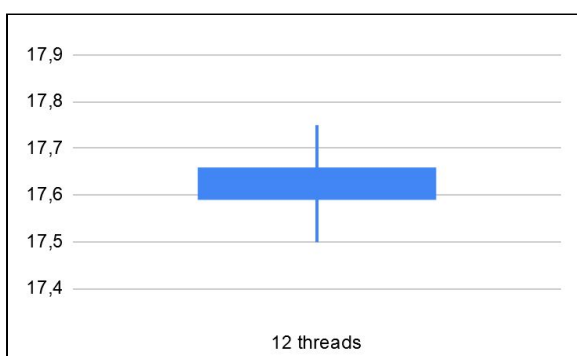
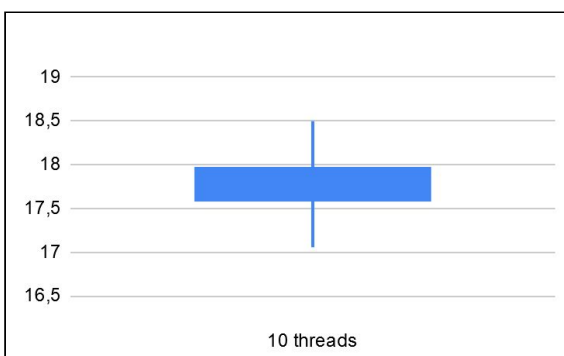
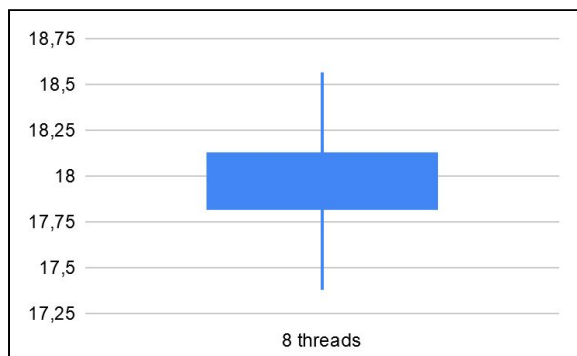
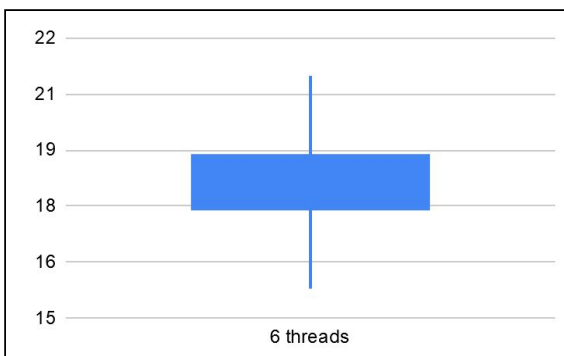
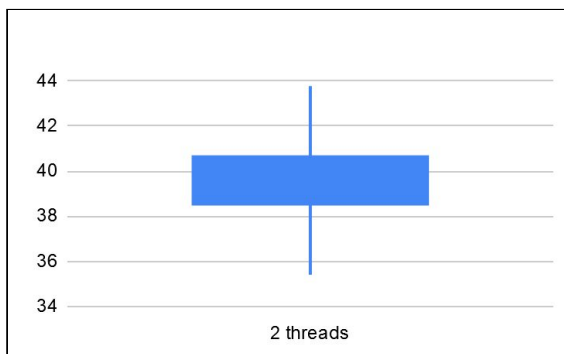
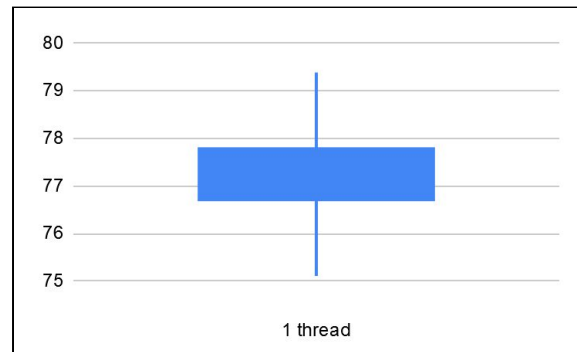
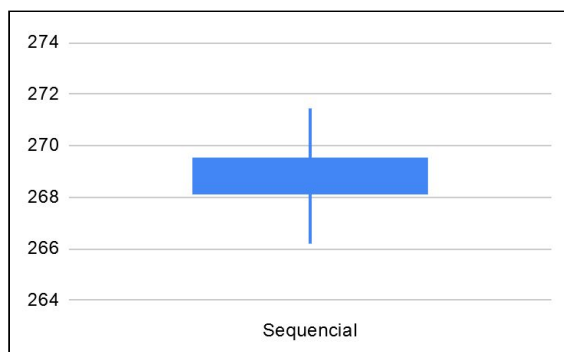
Giorgio Rossa e Felipe Lopes

Pelotas, 13 de Novembro de 2020

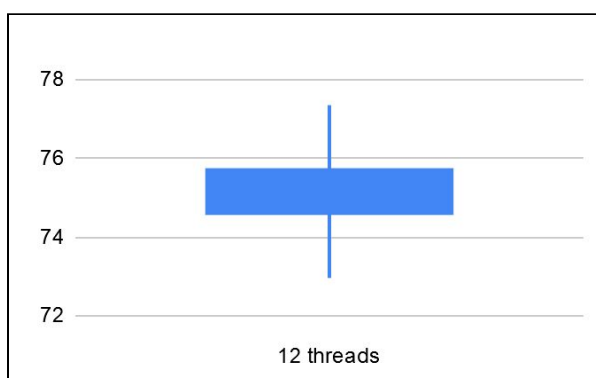
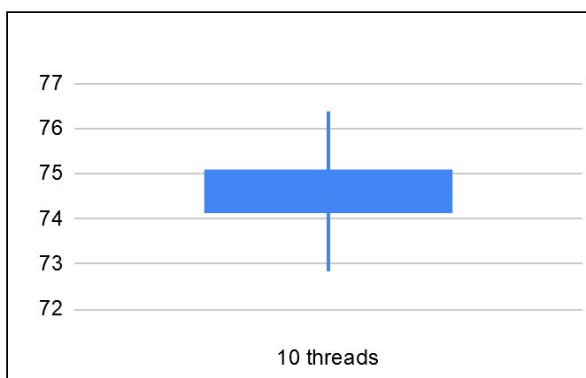
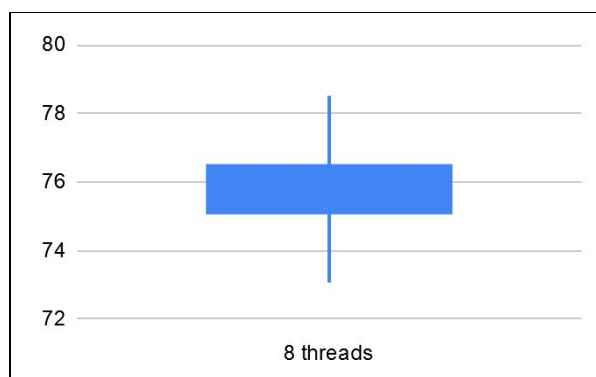
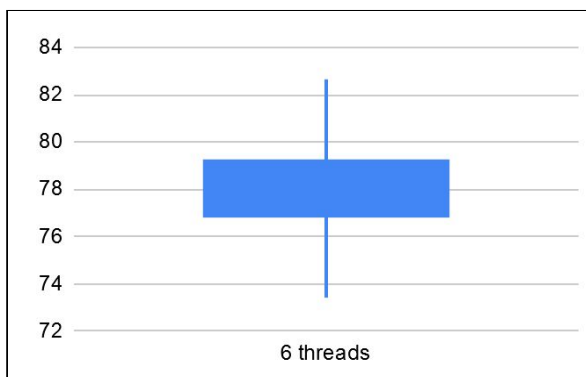
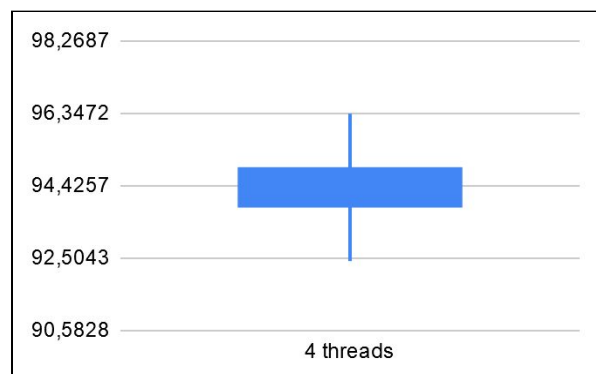
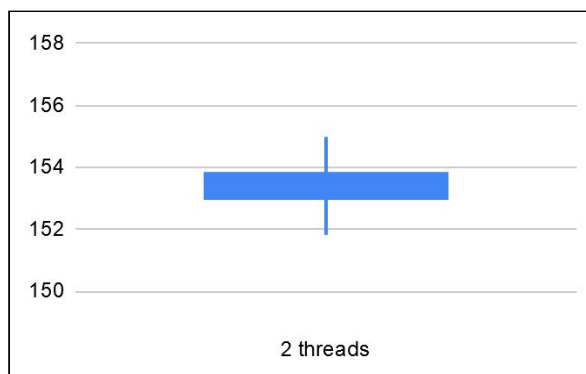
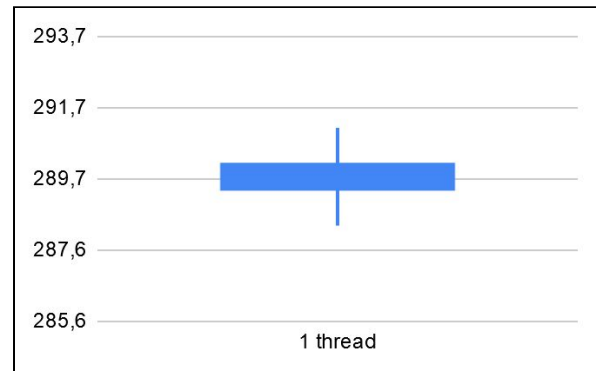
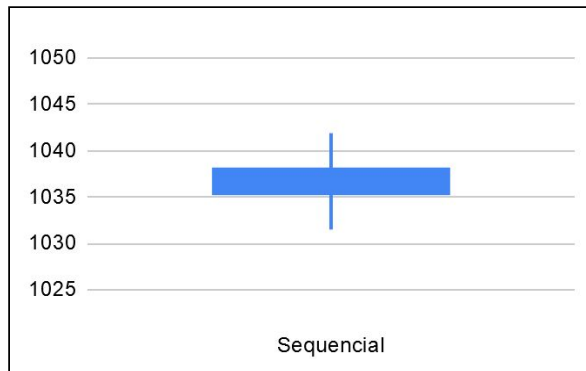
Em nossa análise de desempenho, utilizamos dois ambientes diferentes. Em cada ambiente foram feitos os mesmos testes, variando o número de threads e tamanho da entrada. Cada teste específico foi executado trinta vezes com cem iterações cada, e o valor de tempo foi a média das trinta execuções. Os parâmetros de compilação utilizados em cada ambiente foram `-fopenmp -O3 -std=c++11 -lpthread` para a opção paralela e `-O3 -std=c++11` para a versão sequencial.

O ambiente de avaliação 1 é composto por um processador Fx 6300, com clock de 3,5 GHz e 6 cores físicos. O sistema operacional do ambiente de avaliação 1 é Ubuntu, versão 20.04. A versão do GCC utilizada foi a 9.1.

Os gráficos boxplot para cada quantidade de threads com o tamanho de entrada pequeno (16384) no ambiente 1 são apresentados abaixo. Na vertical temos os tempos coletados em segundos.

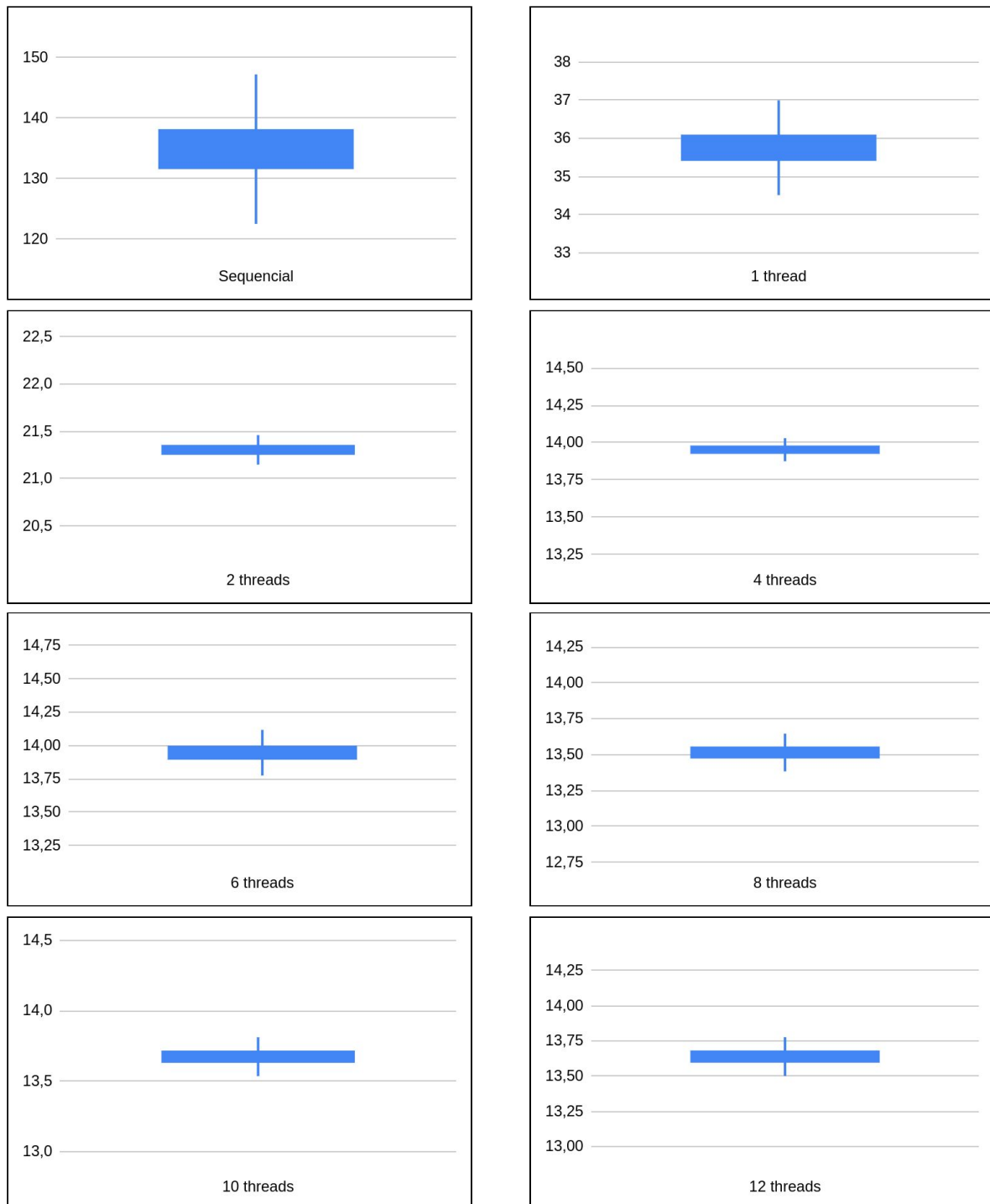


Os gráficos boxplot para cada quantidade de threads com o tamanho de entrada grande (32768) no ambiente 1 são apresentados abaixo. Na vertical temos os tempos coletados em segundos.

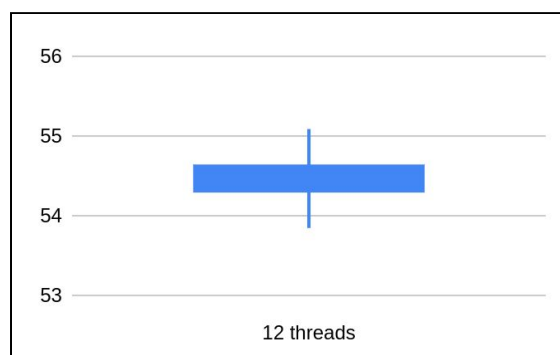
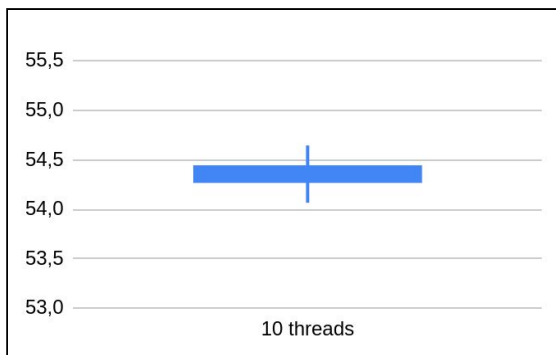
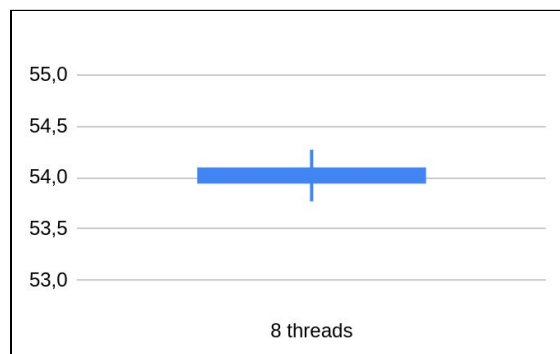
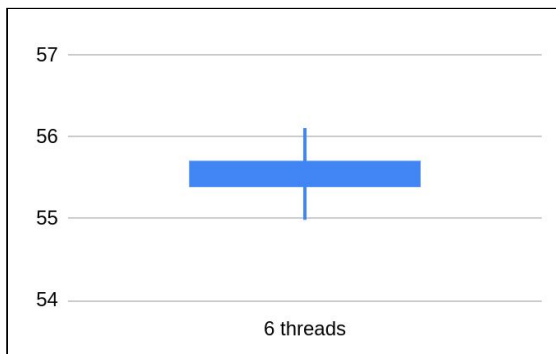
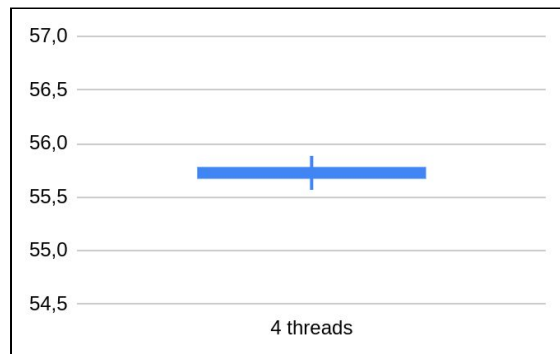
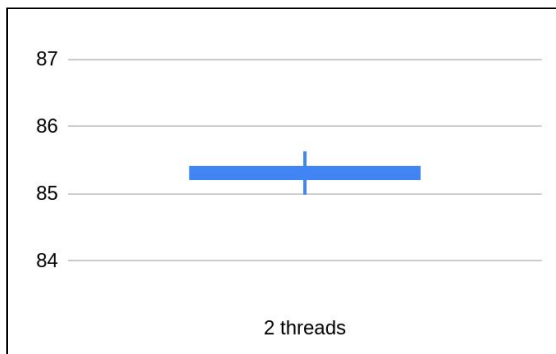
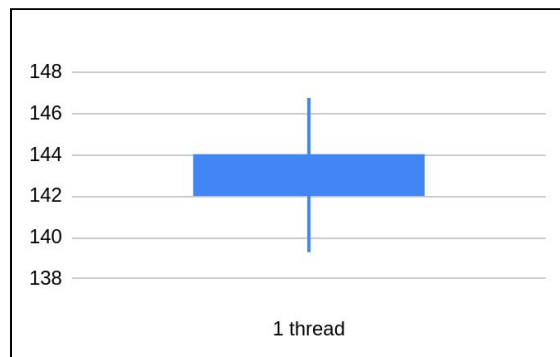
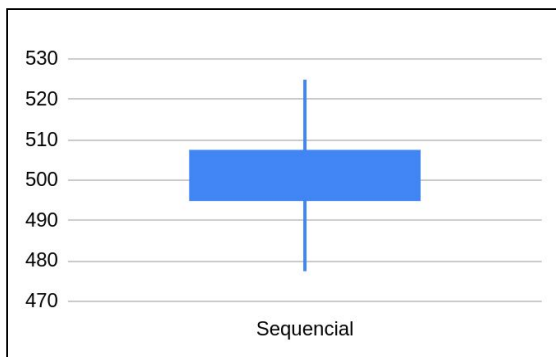


O ambiente de avaliação 2 é composto por um processador Intel Core, modelo i5-8265U, com clock de 1,60 GHz, 4 cores físicos e 8 lógicos. O ambiente de avaliação 2 utiliza o sistema operacional Linux Mint, versão 19. A tabela 2 contém os resultados obtidos neste ambiente. A versão do GCC utilizada foi a 7.5.

Os gráficos boxplot para cada quantidade de threads com o tamanho de entrada pequeno (16384) no ambiente 2 são apresentados abaixo.



Os gráficos boxplot para cada quantidade de threads com o tamanho de entrada grande (32768) no ambiente 2 são apresentados abaixo.



Tempo médios

Os tempos médios estão dispostos em duas tabelas e em dois gráficos, onde é possível visualizar as médias obtidas pelos dois ambientes utilizados durante os testes. As células em vermelho representam médias que não aderem a curva normal, assim não podendo ser consideradas representativas.

Média dos tempos de execução para quantidade de threads								
Ambiente	Sequencial	1 thread	2 threads	4 threads	6 threads	8 threads	10 threads	12 threads
1	268,529763	77,082302	39,382346	24,480149	18,454387	17,928443	17,666571	17,623404
2	135,849759	35,89792	21,3134085	13,9515875	13,9327985	13,515602	13,6593965	13,6356885

Tabela 1. Média em segundos obtida nos dois ambientes para entrada pequena.

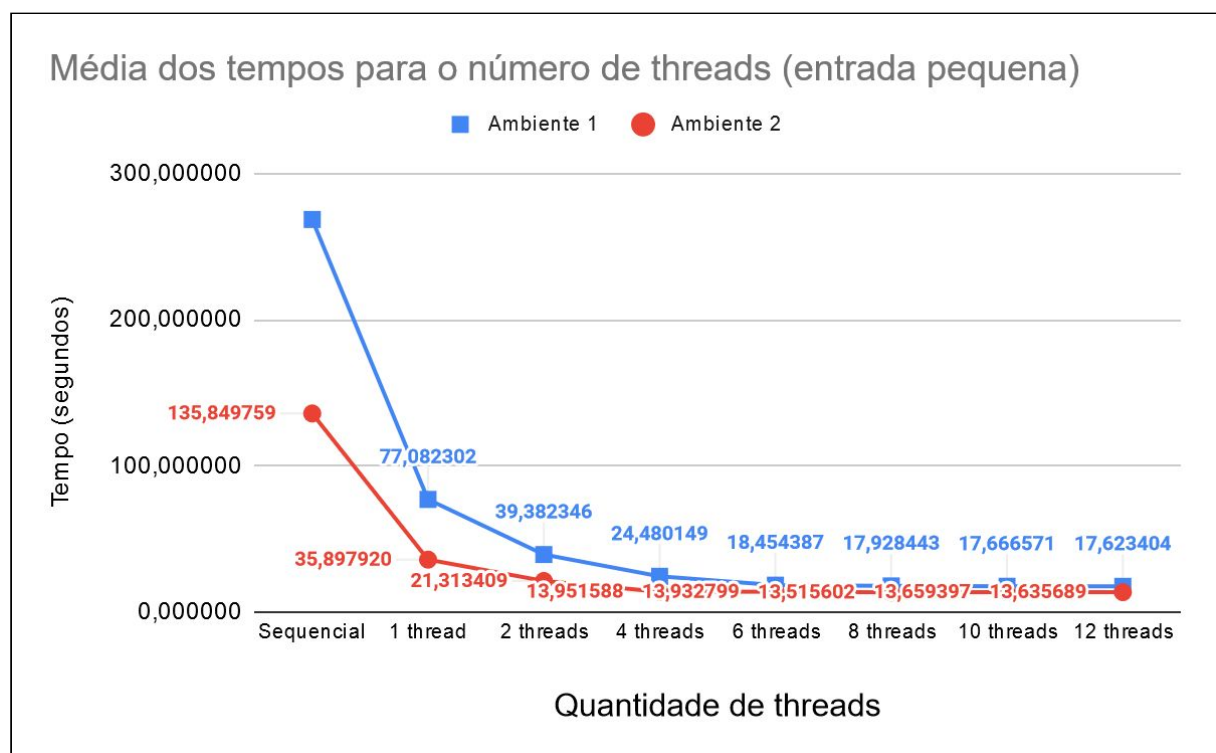


Gráfico 1. Comparativo entre as médias nos ambientes de teste.

Média dos tempos de execução para quantidade de threads								
Ambiente	Sequencial	1 thread	2 threads	4 threads	6 threads	8 threads	10 threads	12 threads
1	1.035,859159	289,591246	153,129364	94,426607	78,216771	76,173217	74,540616	75,125068
2	496,038748	143,829603	85,3198775	55,7154725	55,527271	54,028176	54,3411165	54,445034

Tabela 2. Média em segundos obtida nos dois ambientes para entrada grande.

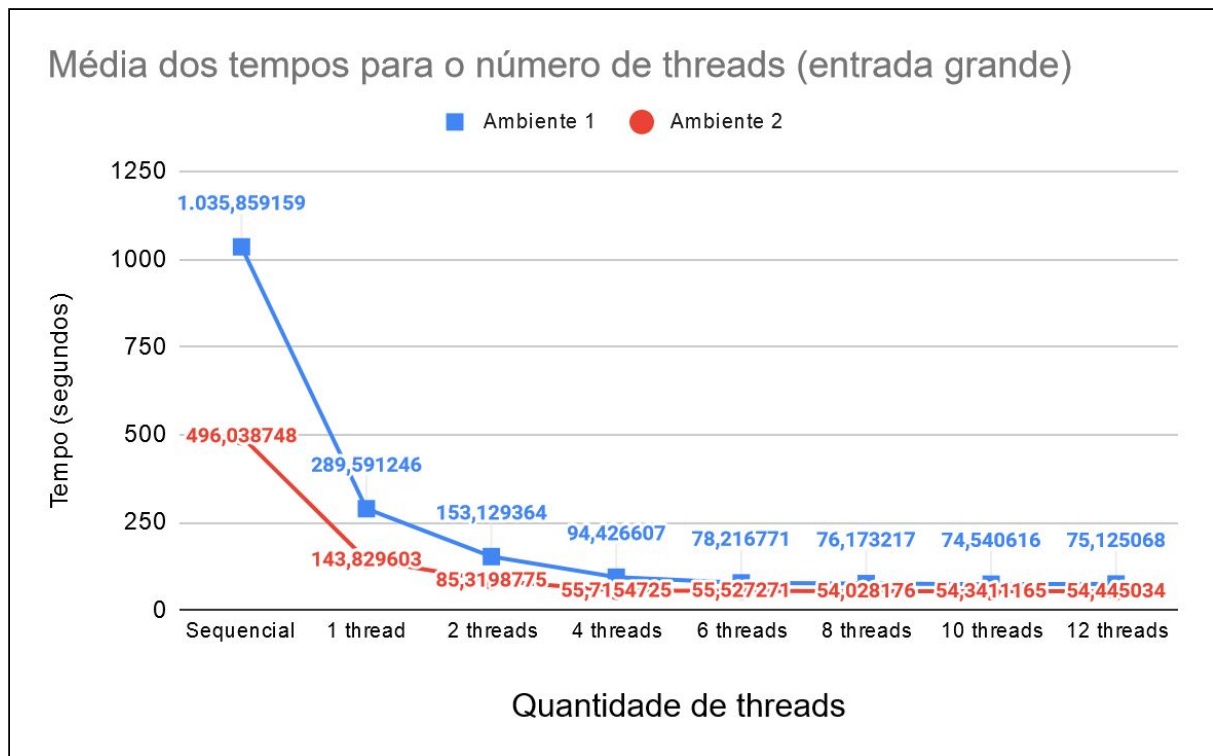


Gráfico 2. Comparativo entre as médias nos ambientes de testes.

Testes de hipótese

Sejam M_1 a média de tempo da execução sequencial das amostras para o tamanho de entrada pequeno e M_2 a média de tempo de execução das amostras do programa paralelo com 1 thread para o tamanho de entrada pequeno,

$$H_0: M_1 = M_2$$

$$H_1: M_1 > M_2$$

Utilizando o teste T de student, tendo uma variância combinada de aproximadamente 0,45733 no ambiente 1 e 1,0820 no ambiente 2, nível de significância de 0,05 e grau de liberdade 58, têm-se o valor do cálculo da seguinte forma:

$$t_0 = (M_1 - M_2) / \sqrt{s_{1,2}^2 * (1/N_1 + 1/N_2)},$$

onde $s_{1,2}^2$ é a variância combinada das amostras sequenciais e com 1 thread e N_1 e N_2 são os tamanhos das amostras.

Desta forma $t_0 = 4246,44593$ para o ambiente 1 e $t_0 = 1441,320966$ para o ambiente 2.

Com o grau de liberdade 58 e nível de significância de 0,05, o valor crítico do teste bilateral é de 2,0017 (positivo ou negativo).

Nos 2 casos, t_0 ultrapassou o valor crítico, sendo assim a hipótese nula (H_0) foi rejeitada. Além disso, é possível afirmar que, pelos dois resultados serem positivos, o valor de M_1 (média dos tempos sequenciais para a entrada pequena) é maior que o de M_2 , ou seja, a média de tempo para a execução do programa sequencial com a entrada pequena é maior que a do programa com uma thread para a entrada pequena.

Utilizando o tamanho de entrada grande, nenhum dos 2 ambientes obteve uma média de tempo de execução representativa, pois os valores não aderem à uma curva normal.

Através dos resultados obtidos parece plausível pressupor que o programa utilizando 12 threads tem melhor desempenho. Para provar isso utilizamos o teste t de student com a hipótese que o programa com 8 threads tem o mesmo desempenho que o programa com 12 threads. Foi escolhido o programa com 8 threads pois, para o tamanho de entrada pequeno foi o que obteve melhor resultado no ambiente de avaliação 2 e o segundo melhor no ambiente de avaliação 1, tendo em vista que a média para 10 threads não foi representativa.

Assim sendo, seguem abaixo os testes de hipóteses referentes às entradas pequena e grande em ordem.

Sendo M_1 a média de tempo de execução para o programa com 8 threads e M_2 a média de tempo de execução para o programa com 12 threads, as duas para a entrada de tamanho pequeno.

$$H_0: M_1 = M_2$$

$$H_1: M_1 \neq M_2$$

Utilizando o teste T de student, tendo uma variância combinada de aproximadamente 0,04025 no ambiente 1 e 0,01471 no ambiente 2, nível de significância de 0,05 e grau de liberdade 58, têm-se o valor obtido como $t_0 = 22,80666546$ no ambiente 1 e $t_0 = -14,85251719$ no ambiente 2. O valor crítico continua sendo 2,0017 (positivo ou negativo), o que significa que a hipótese nula é rejeitada, ou seja, as médias são diferentes. No ambiente 1, o valor encontrado é positivo, o que significa que a menor média de tempo é com 12 threads, enquanto no ambiente 2 o valor encontrado é negativo, significando que a menor média de tempo é com 8 threads.

Sendo M_1 a média de tempo de execução para o programa com 8 threads e M_2 a média de tempo de execução para o programa com 12 threads, as duas para a entrada de tamanho grande.

$$H_0: M_1 = M_2$$

$$H_1: M_1 \neq M_2$$

Utilizando o teste T de student, tendo uma variância combinada de aproximadamente 0,24633 no ambiente 1 e 0,044008 no ambiente 2, nível de significância de 0,05 e grau de liberdade 58, têm-se o valor obtido como $t_0 = 31,67779756$ no ambiente 1 e $t_0 = -29,80643351$ no ambiente 2. O valor crítico continua sendo 2,0017 (positivo ou negativo), o que significa que a hipótese nula é rejeitada, ou seja, as médias são diferentes. No ambiente 1, novamente o valor foi positivo, significando que a menor média de tempo é com 12 threads, enquanto que no ambiente 2, outra vez o valor encontrado foi negativo, o que evidencia que a menor média de tempo é com 8 threads.

Diante dos resultados obtidos é possível supor que o processador do ambiente 1 obteve melhor desempenho com 12 threads por possuir um número de cores múltiplo de 6 e isso ocorre também no ambiente 2, onde o processador utilizado possui cores múltiplos de 4, obtendo um desempenho superior com 8 threads. Acreditamos que essa multiplicidade

pode ser impactante para o resultado final, pois facilitaria o trabalho do escalonador, gerando um overhead menor. Além disso, o resultado com 8 threads obtido no ambiente 2 já era pequeno, e o custo da troca de contexto entre threads acabou impactando mais nesse cenário conforme se aumentou o número de threads em uso, isto é visível quando analisamos o gráfico 1, onde desde o número de threads 4 não existe grande diminuição ou aumento de tempo.

Conclusão

Dados os resultados obtidos ao longo de nossos testes, podemos notar que existe um ganho com a adição de threads. Este ganho se evidencia com a diferença entre a execução do programa de forma sequencial e com 2 threads, porém no ambiente 1 uma melhora continuou acontecendo até o último teste com 12 threads, e no ambiente 2 até 8 threads. Como discutido nos testes de hipótese, isto se deve à fatores relacionados a arquitetura de cada ambiente de avaliação utilizado.

No ambiente 1 existe vantagem em tempo de execução para utilizar-se um maior número de threads, enquanto no ambiente 2 a vantagem só existe até certo ponto. Isto pode acontecer tanto pela arquitetura em número de cores físicos e lógicos, como por questões do escalonador, sistema operacional e a forma de implementação do processador, pois os dois ambientes possuem processadores bem diferentes.

Assim, podemos concluir que implementar programas paralelos pode ser um desafio se não se conhecer bem o ambiente que se está utilizando, pois é natural pensar que mais é mais, mas não nesse caso, nossos resultados demonstram que dependendo do ambiente em que se está executando o programa menos pode ser mais, o que ficou demonstrado pelos testes realizados em nosso ambiente 2. Sendo assim, na implementação de programas paralelos o programador deve levar fortemente em consideração o ambiente no qual esse programa irá ser executado, pois evidenciamos que uma solução ótima para o ambiente 1 não seria ótima para o ambiente 2 e vice versa.

Todos os testes, tabelas e gráficos feitos foram produzidos em uma planilha que encontra-se disponível [aqui](#). A planilha não possui explicações, apenas os dados brutos dos testes.